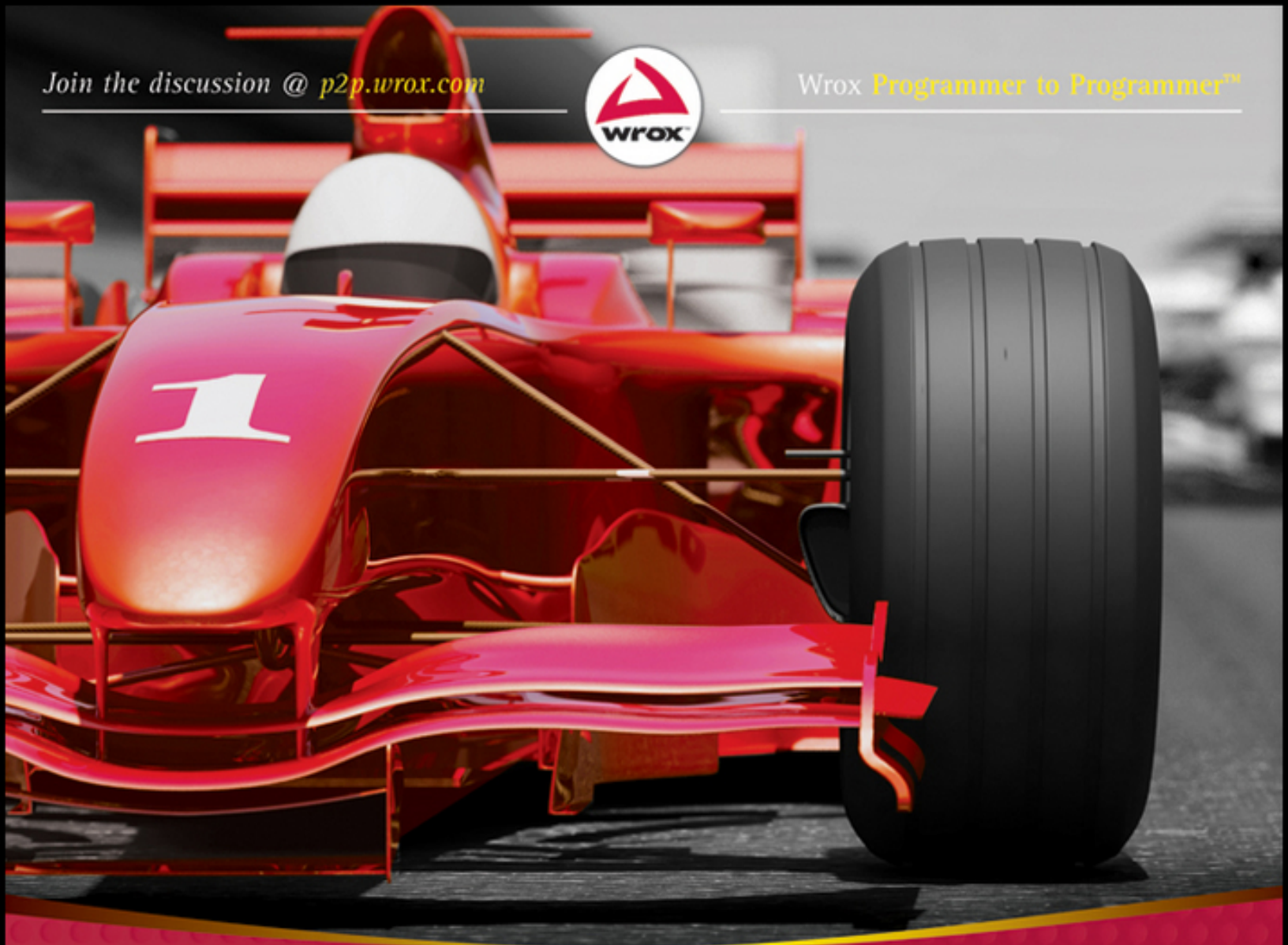


Join the discussion @ [p2p.wrox.com](http://p2p.wrox.com)



Wrox Programmer to Programmer™



Professional

Microsoft®

# SQL Server® 2012 Reporting Services

Foreword by Ariel Netz, *Partner Group Program Manager for Microsoft SQL Server Reporting Services*

Paul Turley, Robert Bruckner, Thiago Silva, Ken Withee, Grant Paisley

# Table of Contents

## Part I: Getting Started

### Chapter 1: Introducing Reporting Services

Who Uses Reporting Services?

Dashboards, Reports, and Applications

Enterprise Reporting

Solution Types

Business Intelligence Reporting Solutions

Customizing the Reporting Experience

Summary

### Chapter 2: Reporting Services Installation and Architecture

The Basic Installation

The Enterprise Deployment

The Reporting Life Cycle

Reporting Services Tools

Reporting Services Windows Service

Reporting Services Processors and Extensions

Reporting Services Application Databases

Summary

### Chapter 3: Configuring SharePoint Integration

The SharePoint Technologies

Installation and Configuration



[Architecture](#)  
[Summary](#)

## [Part II: Report Design](#)

### [Chapter 4: Basic Report Design](#)

[What Good Are Wizards, Anyway?](#)  
[Manual Report Design](#)  
[Summary](#)

### [Chapter 5: Report Layout and Formatting](#)

[Report Layout Types](#)  
[Designing Multicolumn Reports](#)  
[Designing Gauge Reports](#)  
[Summary](#)

### [Chapter 6: Designing Data Access](#)

[Business Intelligence Reporting](#)  
[Reporting for Relational Data](#)  
[Filtering Techniques](#)  
[Using Other Data Sources](#)  
[Best Practices](#)  
[Summary](#)

### [Chapter 7: Advanced Report Design](#)

[Headers and Footers](#)  
[Creating Report Templates](#)  
[Creating Composite Reports](#)  
[Designing Subreports](#)

[Navigating Reports](#)  
[Reporting on Recursive Relationships](#)  
[Summary](#)

## [Chapter 8: Chart Reports](#)

[Chart Types](#)  
[Creating a Multiseries Chart](#)  
[Summary](#)

# [Part III: Business Intelligence Reporting](#)

## [Chapter 9: BI Semantic Models](#)

[Introduction to Data Modeling](#)  
[The BI Semantic Model](#)  
[Summary](#)

## [Chapter 10: Reporting with Analysis Services](#)

[Why Use Analysis Services for Reporting?](#)  
[Using Reporting Services with Analysis Services Data](#)  
[Working with Multidimensional Expression Language](#)  
[Adding Nonadditive Measures](#)  
[MDX Properties and Cube Formatting](#)  
[Drill-Through Reports](#)  
[Best Practices and Provisions](#)

[Summary](#)

## [Chapter 11: OLAP Reporting Advanced Techniques](#)

[Cube Dynamic Rows](#)

[Cube Dynamic Rows Expanded](#)

[Cube Restricting Rows](#)

[Cube Metadata](#)

[Cube Browser](#)

[Summary](#)

## [Part IV: Enabling User Reporting](#)

### [Chapter 12: Tabular Models](#)

[Introduction to PowerPivot](#)

[Importing Data into PowerPivot](#)

[PowerPivot Window](#)

[Analyzing and Enriching Data](#)

[Summary](#)

### [Chapter 13: Visual Analytics with Power View](#)

[Introduction to Power View](#)

[Power View Architecture](#)

[Preparing a Model and Connection for a Tutorial](#)

[Visual Analytics with Power View](#)

[Configuring Data Source Connections](#)

[Analysis Services Tabular](#)



[Summary](#)

## [Chapter 14: Report Builder Solution Strategies](#)

[Report Builder and Semantic Model History](#)

[Planning a Self-Service Reporting Environment](#)

[User Report Migration Strategies](#)

[Summary](#)

## [Part V: Solution Patterns](#)

### [Chapter 15: Managing Report Projects](#)

[Solutions and Projects](#)

[Version Control](#)

[Synchronizing Content](#)

[Managing Server Content](#)

[Getting Started with Azure Reporting](#)

[Summary](#)

### [Chapter 16: Report Solutions, Patterns, and Recipes](#)

[Super Reports](#)

[Report Recipes: Building on Basic Skills](#)

[Summary](#)

## [Part VI: Administering Reporting Services](#)

## Chapter 17: Content Management

Using Report Manager

Content Management Activities

Item-Level Security

Content Management Automation

Summary

## Chapter 18: Integrating Reports with SharePoint

Native Mode Web Parts

Publishing Reports to SharePoint

Report Management

Integrated Mode Web Parts

Native Mode Versus Integrated Mode

Report Models

Summary

## Chapter 19: Native Mode Server Administration

Security

Backup and Recovery

Monitoring

Configuration

Summary

## Part VII: Reporting Services Custom Programming

## [Chapter 20: Integrating Reports into Custom Applications](#)

[URL Access](#)

[Programmatic Rendering](#)

[Using the ReportViewer Control](#)

[Summary](#)

## [Chapter 21: Using Embedded and Referenced Code](#)

[Using the Expression Builder](#)

[Calculated Fields](#)

[Conditional Expressions](#)

[IIF\(\) Is Your Friend](#)

[Using Custom Code](#)

[Summary](#)

## [Chapter 22: Extending Reporting Services](#)

[Extension Through Interfaces](#)

[Creating a Custom Data Processing Extension](#)

[Summary](#)

## [Part VIII: Appendixes](#)

### [Appendix A: T-SQL Command Syntax Reference](#)

[T-SQL Commands, Clauses, and Predicates](#)

[Script Comment Conventions](#)

[Reserved Words](#)



## [Appendix B: T-SQL System Variables and Functions](#)

[System Global Variables](#)

[System Functions](#)

## [Appendix C: MDX Reference](#)

[Object Identifiers](#)

[Reserved Keywords](#)

[Member References](#)

[Sets](#)

[Tuples](#)

[The SELECT Statement](#)

[MDX Functions and Keywords](#)

## [Foreword](#)

## [Introduction](#)

## [Advertisements](#)

# Part I

## Getting Started

- Chapter 1: Introducing Reporting Services
- Chapter 2: Reporting Services Installation and Architecture
- Chapter 3: Configuring SharePoint Integration

# Chapter 1

## Introducing Reporting Services

What's in this chapter?

- Understanding report designer roles and the tools used to design reports

- Understanding dashboards, reports and applications

- Examining Business Intelligence solutions

- Discovering multidimensional and tabular semantic models

You're holding this book, trying to decide if it will help you solve a problem or teach you essential skills to create reports with Reporting Services. If you and I were having this conversation in person, I'd ask you to tell me what you need. I teach classes and travel to companies to create report and BI solutions, and at the beginning of every class or consulting engagement, I ask what the student or client needs. What are the requirements? What questions does your report need to answer? What's not working? What needs to be fixed, and what will it take to build a solution to help you reach your goals? So, I ask you, what do you need? Why are you reading a book about Reporting Services? Do you have a specific problem to resolve, or do you just need to develop some basic report design skills? Do you need to build an entire reporting solution? Who are the users of these reports? Are they department workers, business managers, or financial analysts? Maybe your user is the CEO of a major corporation or other business executives who need to know if the company is on the right track. Maybe you need to create reports for your own business to make



sure it's profitable and achieving its goals. Whether you are creating an invoice to sell arts and crafts out of your garage or a BI dashboard to help manage a multinational corporation, the reports you will create are important. Therefore, you need to make sure they deliver accurate information and are designed correctly, using industry best practices. That's a big responsibility.

Whatever your needs, we'll cover all these bases and address each topic thoroughly. I've enlisted some of my most trusted associates to share their experiences.

This chapter is a high-level introduction to the concepts and capabilities of this powerful reporting tool and the data analysis platform of Microsoft SQL Server 2012. It introduces common reporting scenarios, beginning with the most basic and then moving to the more advanced. In subsequent chapters, you will explore these capabilities in depth and learn how to use them in your own reporting solutions.

SQL Server Reporting Services has grown to become the de facto industry standard reporting tool by which others are measured. It is a foundation upon which you can construct complete report, scorecard, and dashboard solutions for business users. Today, it does everything from simple ad hoc data reporting to delivering enterprise-ready integrated reporting into business portals and custom applications.

Not long ago, the information technology (IT) group for a large financial services company wanted to make sure that they were using the best reporting tool on the market. They decided to hire a consulting company to evaluate every major reporting product and give them an unbiased analysis. I was lucky to land this assignment. We worked with the client to identify about 50 points of evaluation criteria. Then I contacted all the major vendors, installed evaluation copies and explored features, and spoke with other customers and with those who specialized in using

these various products. It really helped us see the industry from a broad perspective and was a valuable learning experience. There are some respectable products on the market, and all have their strengths, but I can honestly say that Microsoft has a unique and special platform. As a consultant, contractor to Microsoft, and Microsoft SQL Server MVP, I have had the opportunity to work alongside the Reporting Services product team for many years. They have a vision, and they're passionate about their product. I have a great deal of respect for the fine people who continue to develop and improve Reporting Services, version after version.

## **Who Uses Reporting Services?**

Business users fit into a few categories when we consider how they use reports. Some are report consumers only. They're happy to use reports that have been written and published for them. Others prefer to create their own reports using business tools they understand and use for other things, as Excel is used for planning and financial analysis. Maybe they just want to browse information to look for trends and to understand how the business is measuring up against their goals. Still other business users want to use more sophisticated tools to create powerful reports. A typical information technology group at most large organizations has three common roles: system administrators, application developers, and project managers. Usually everyone else in the department supports these roles. Where does the report designer fit in the organization? Good question. Honestly, I don't have a simple answer. The fact is that people who design business reports don't come from a common pool of IT professionals.

In fact, many people who spend the majority of their time creating reports are part of the business community and are not your typical hard-core computer geeks.

Microsoft has a long history of building highly technical products that appeal to the technical community. In more recent years, Microsoft has begun to enhance its product culture for more suit-and-tie-wearing folks who talk about things such as business performance management strategy and market share rather than remote procedure calls and polymorphic object inheritance. If you're a business-type person, you probably don't care about integrating your reports into custom applications and web sites or writing complex programming logic to make them sing and dance. Some of us live for that. What you may care about is giving your savvy business user the ability to drag and drop report parts from the gallery to visualize important key metrics to see what products are performing well in their sales territory. However, to enable that experience, a certain degree of technical expertise is necessary.

Over the years, I've taken inventory of the people who consider themselves report designers. They generally fall into one of two camps — business-focused or technology-focused. There has been a significant shift toward more accessible reporting tools for those who have less technical aptitude. The following roles represent the majority and describe some of the trends we're seeing as the industry continues to evolve.

## **Business Information Workers**

People in this role have strong computer skills, but they don't spend their time writing code and using programming tools. Their primary interest is exploring information and finding answers, rather than designing complex reports. If you're an information worker (IW), you need easy-to-use tools to browse data and create simple reports quickly and



with less technical expertise. IWs typically create a report to answer a specific question or address a particular need, and then they may discard the report or save it to a personal area for reuse. They tend to create a separate report for each task and may or may not share these reports with others who have similar needs. This is by far the largest and fastest-growing group of report tool users in the industry.

## **Business Managers**

If you're a business manager, you're primarily interested in your own domain of the business. Managers need reports to support specific processes to address their analytical needs and to help them make informed decisions. Like information workers, they have little interest in the implementation details or technology used to make it work. As information workers, managers may create their own reports to analyze the productivity of their team or area of responsibility.

## **Software Developers**

To achieve advanced reporting features, software developers write complex queries and custom programming code to process business rules and give reports conditional formatting and behavior. Developers typically feel right at home with the report design environment because it's similar to familiar programming tools. However, report design is not the same as application development. Designing a report can be faster and easier in some ways than developing software. Advanced report design can involve writing code and even developing custom components.

## **System Administrators**

If you are a system administrator, you are typically concerned with the setup and ongoing maintenance of servers and the infrastructure to keep reporting solutions available and working. Administrators typically spend their time and energy managing security and optimizing the system for efficiency. Reporting Services has an administrative component that is especially important in large-scale implementations.

In smaller organizations, the same person may play the role of system administrator, developer, and report designer. Reports can also be created to help monitor system usage and maintenance statistics to make a system administrator's job easier.

Reporting Services meets the needs of information workers and technology professionals with different report design and data browsing tools. Report Builder is simple, focused, and familiar, with a user interface similar to Microsoft Office applications. A more advanced report designer, called SQL Server Data Tools (SSDT), is available to application developers and other technical professionals. It is integrated into the Visual Studio solution design environment shell.

## **Dashboards, Reports, and Applications**

From a software perspective, a computer system can present data to a user in different ways. Most legacy reporting systems ran on the client desktop computer.

Just recently we've seen a major shift toward self-service reporting. Many different tools have been developed to provide the right balance between simplicity and capability. It's taken a few years for the dust to settle on all these tools and for the best options to emerge. Most business leaders

want a simple dashboard interface that answers key business questions, or a tool they can use to browse a simplified view or semantic model without having to design reports and write queries. This led Microsoft to take two different approaches. First, it created a streamlined report design tool, Report Builder, that makes it easier for less-technical users to design powerful reports using the conventional report definition architecture. Second, it developed a separate visualization tool, Power View, that leverages the Reporting Services architecture but offers a design and user experience that is separate from conventional reports. To manage expectations, we'll omit the Power View tool from our discussions and treat it as separate from the rest of Reporting Services. You'll read about tabular models and Power View in Chapters 12 and 13.

Quite a few years ago we saw a shift from client-based processing toward applications that ran on web servers. This has proven to be an effective way to make systems available to a large number of people. When Reporting Services was first released, it was available only as a server-based solution, with reports delivered almost exclusively through the web browser — and this is primarily how SQL Server Reporting Services (SSRS) reports are used today. However, the capabilities don't stop there. Reporting Services lets you run reports in a variety of modes and applications. If we've learned anything from the past 20 years of computer system evolution, it's that centralized server-based solutions and client-side applications each offer unique advantages and trade-offs in terms of features, capabilities, interactive user experience, and scalability.

It's important to note that you can install the product and then design and use reports without a lot of fuss and technical expertise. Later in this book, we will discuss how

Reporting Services can be used in more advanced and creative ways.

## **Blurring the Line Between Applications and Reports**

With Reporting Services, you can integrate reports into applications in such a way that users may not be able to tell the difference between the report content and the application interface. With a little bit of programming code, reporting features can be extended to look and act much like applications. When do reports start replacing application functionality? What, exactly, is the difference between a report, a dashboard, and a scorecard? The lines have become quite blurred. Your task is to decide which tool best meets your needs. Many intranet sites run on web portals, rather than custom-programmed web sites, and Reporting Services naturally plays well in practically any web portal environment. In particular, Reporting Services has native integration with Microsoft SharePoint Server.

The exciting news is that you now have a tool that can do some incredible things. As my favorite superhero's uncle said, "With great power comes great responsibility." If you are a simple report designer with simple needs, the good news is that using Reporting Services to design simple reports is, well, simple. If you are a software developer and you intend to use this powerful framework to explore the vast reaches of this impressive technology, welcome to the wonderful world of creative custom reporting.

After years of experience with this product, I've learned an important lesson on this topic. They say that to a hammer, everything looks like a nail. Likewise, to a programmer, a lot of challenges may look like an opportunity to write program code. That may be the right solution under certain conditions. But often, the most effective solution is to simply

use a feature already baked into the product — and implement that feature as it was designed to work. I often have this conversation with programmers after they have spent hours writing a complicated solution to a simple problem. My father often advised me to follow the well-known KISS principle: “Keep it simple, stupid.” The last part was just to make the phrase memorable, and he meant it in the most affectionate way (at least, I'm pretty sure he did).

## **Launching Reports from an Application**

Hyperlinks and application shortcuts can easily be added to documents and custom applications. Much of the standard report-viewing environment may be controlled using parameters passed to the report server in the URL. Reports may be designed to prompt users for parameter values used to filter data and to modify the report format and output. These parameters may also be incorporated into a URL string. This way, one hyperlink displays a report with one set of data, and another hyperlink displays the same report with different data. Parameters can even be used to change display attributes such as font sizes and colors, and to hide and show content.

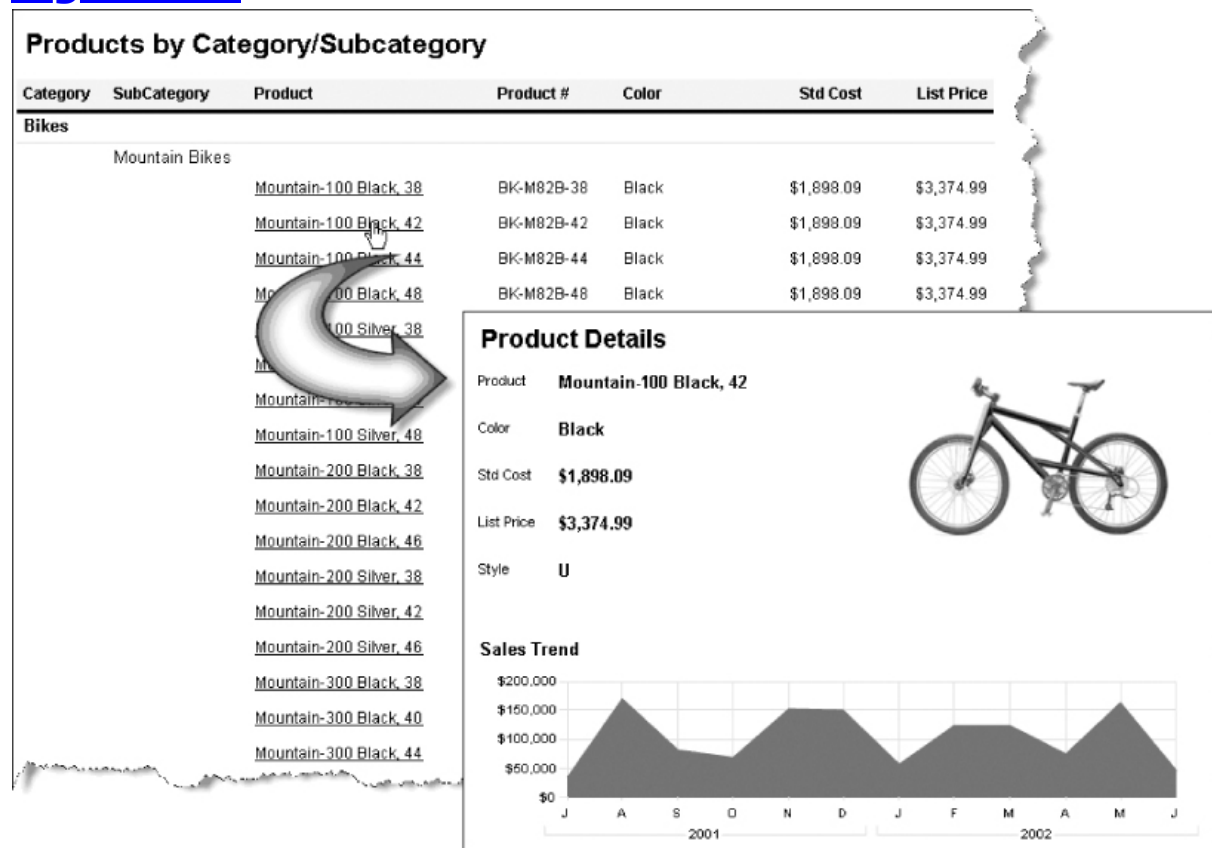
## **User Interaction**

In the past, many reports were nothing more than a list of values with totals. Now reports can be a starting point that can guide users to the information they need to make decisions. Report elements such as text labels, column headers, and chart data points can be used to navigate to different report sections and new reports. Because navigation links may be data-driven and dynamically created based on program logic, report links may also be used to navigate into business applications. Imagine using

your reports to launch programs and to navigate to document libraries and online content!

As the user clicks items or data points in the report, content and the layout of a report can change based on parameter values set in the background. Summary headers may be used to expand and collapse detail sections, giving users the ability to drill down to more specific information, as shown in [Figure 1.1](#).

**[Figure 1.1](#)**



## Integrating Reports and Applications

One of the reasons that Reporting Services integrates so easily with modern web applications is that it natively supports Hypertext Markup Language (HTML), the standard markup language used to create web pages. Techniques



may be used to incorporate reports into a web application in a variety of ways:

- Hyperlinking to navigate the web browser window to a report
- Hyperlinking to open reports in a separate web browser window, with control over report display and browser features
- Embedding reports into a page using a frame, IFrame, or ReportViewer web control
- Programmatically feeding report content to an Active Server Page (ASP or ASPX) using server-side custom code
- Programmatically writing reports to files available for download from a web site
- Using a web part to embed reports into a SharePoint Web portal
- Fully integrating the report server in SharePoint Integration mode

There are a lot of creative ways to integrate reports into a web or desktop application. These techniques range from simple, requiring a little HTML script, to complex, custom methods. And if it's not enough to be able to embed reports into custom web pages, it's also possible to use custom program code to embed additional content into reports. Imagine the possibilities. Actually, you don't have to imagine anything. Just keep reading!

SharePoint integrated mode allows all your reports and report administration to be managed completely within SharePoint. If you choose to manage the report server separately from your SharePoint portal, you can still use SharePoint web parts to navigate folders and reports and to view reports hosted on the report server running in Reporting Services Native mode.

The ReportViewer control or embedded web browser may be used to view server-based reports in a form. These

reports are still managed on the report server and maintain all the security settings and configuration options defined by an administrator. Queries and data access are still performed on the server. The other option is to embed these reports directly into the client-side application. The Windows Forms ReportViewer controls can act as a lightweight report-rendering engine. This means that reports built into a custom application can run independently from the report server.

## **Enterprise Reporting**

Delivering reports to the masses requires a capable reporting environment. Rather than bringing data from source databases to the desktop for processing, Reporting Services processes queries and then renders reports on the report server. Because it uses Windows Services, shared server-based components, and HTTP web services, all the processing occurs in an efficient and secure environment. Standard data-source connection providers for SQL Server and other enterprise-class databases promote efficient use of server resources. In simple terms, this means that many users can run reports at the same time while consuming minimal server resources.

A business intelligence solution integrates data from multiple sources into a data warehouse, data mart, or semantic data model. Complex analysis solutions often require tabular or multidimensional data structures. If you're not familiar with the terms "OLAP," "tabular," and "semantic model," this might at first be a little confusing. A quick history lesson will clear that up. In the late '90s Microsoft released a product that came with SQL Server 7.0, called "OLAP Services." This was their multidimensional database technology that performed online analytical processing, storing data in cubes and dimensions, rather than tables. In

SQL Server 2000, OLAP Services became Analysis Services. Multidimensional (OLAP) databases store data in a pre-grouped and pre-aggregated format on disk so the data is available quickly for reporting and browsing. In SQL Server 2012, Microsoft rebranded their analytical database technologies as “Business Intelligence Semantic Models” or “BISM.” They offer two technologies under the BISM umbrella; multidimensional (OLAP) and tabular models, which store and process analytic functions in-memory. In some cases, tabular models are easier to design and may be more efficient and faster for reporting and analysis. The chapters in Part III help you understand the advantages of BI data sources and how reports are designed to work with analytic data and semantic models.

The Reporting Services report server exposes its functionality in the same way that a standard ASP.NET web site is hosted for users. Reports may be accessed from anywhere within or outside of the corporate firewall and are still available only to selected users. In SharePoint integrated mode, reports are available to users through document libraries and are secured and managed within the SharePoint server environment. In Native or nonintegrated server mode, reports are managed through a web application called *Report Manager*. Reports can also be exposed in custom-developed web applications using practically any set of web technologies or development tools.

## **Solution Types**

An impressive aspect of Reporting Services is that there are so many different ways to implement reports into a business environment. However, giving people a lot of choices doesn't necessarily solve their problems. In fact, providing users with too many options can just be confusing and

overwhelming. As report system designers, our job is to provide the right kind of solution for our users that is simple, uncluttered, and easy to use. Reporting Services has become such a multifaceted platform that we often must clarify what we mean when talking about “Reporting Services reports.” Part V, “Solution Patterns,” gives you prescriptive guidance about how to create and manage reporting solutions using best practices and solution patterns.

The majority of new Reporting Services implementations for most organizations use the de facto Web-based Report Manager interface or are integrated into a company SharePoint site. Other options to integrate reports into custom applications or web pages may be used to meet specific business needs but are less common. In reality, reports can be integrated into a variety of custom solutions with relative ease. Here are some software solutions that might incorporate reports:

- Out-of-the-box, server-based reporting features, using reports created by report designers and deployed to a central web server.
- Reports integrated into web applications using URL links to open in a web browser window.
- Reports integrated into SharePoint Services applications using SharePoint web parts.
- Custom-built application features that render reports using programming code. Reports can be displayed within a desktop or web application or may be saved to a file for later viewing.
- Interactive data visualizations using the Power View visualization tool for data exposed through a tabular semantic model.

## **Simple Report Design**

If you need to create common report types to summarize or output information contained in a database, Reporting Services offers some great tools that make this easy to do. For example, suppose you have a record of customers and the products they have purchased. You want to produce a list of customers that contains the number of transactions and the total amount the customers have spent. You can use the Report Wizard to produce a table report that includes this information. If you want to compare the sales for each customer, day-to-day, over a period of time, you can use the Report Wizard to generate a line chart report to view the sales trend. The point is that common report types can be easy to create with tools and features that don't require users to know a lot about complicated things like programming, writing queries, and building expressions.

Categorically, report solutions may be created by information technology staff or business users; a variety of tools have been created to support the needs of each group. After a brief explanation, [Table 1.1](#), shown later, summarizes the report designer options we've seen in the current and past product version.

Managing a fully scaled corporate BI solution can be complex and expensive. Fortunately, all the components of a working solution can be scaled down to a single server if necessary. Small and midscale reporting solutions may use a single, multipurpose database serving as an operational data store and a reporting data structure. As the solution matures, the eventual separation of these databases is almost inevitable. A small-scale data mart, populated from operational databases at regular intervals, will provide a simpler data source for reporting that doesn't compete with users and applications for system resources.

Simple reports are easy to design and deploy for short-term use. With a little planning and discipline, you can design reports to meet future requirements. Properly

designed, your reports can include advanced features that meet simple needs now and more sophisticated needs in the future.

The five chapters in Part II, “Report Design,” begin with the fundamentals of basic report design and then progressively demonstrate how to add more advanced functionality.

## IT-Designed Reports

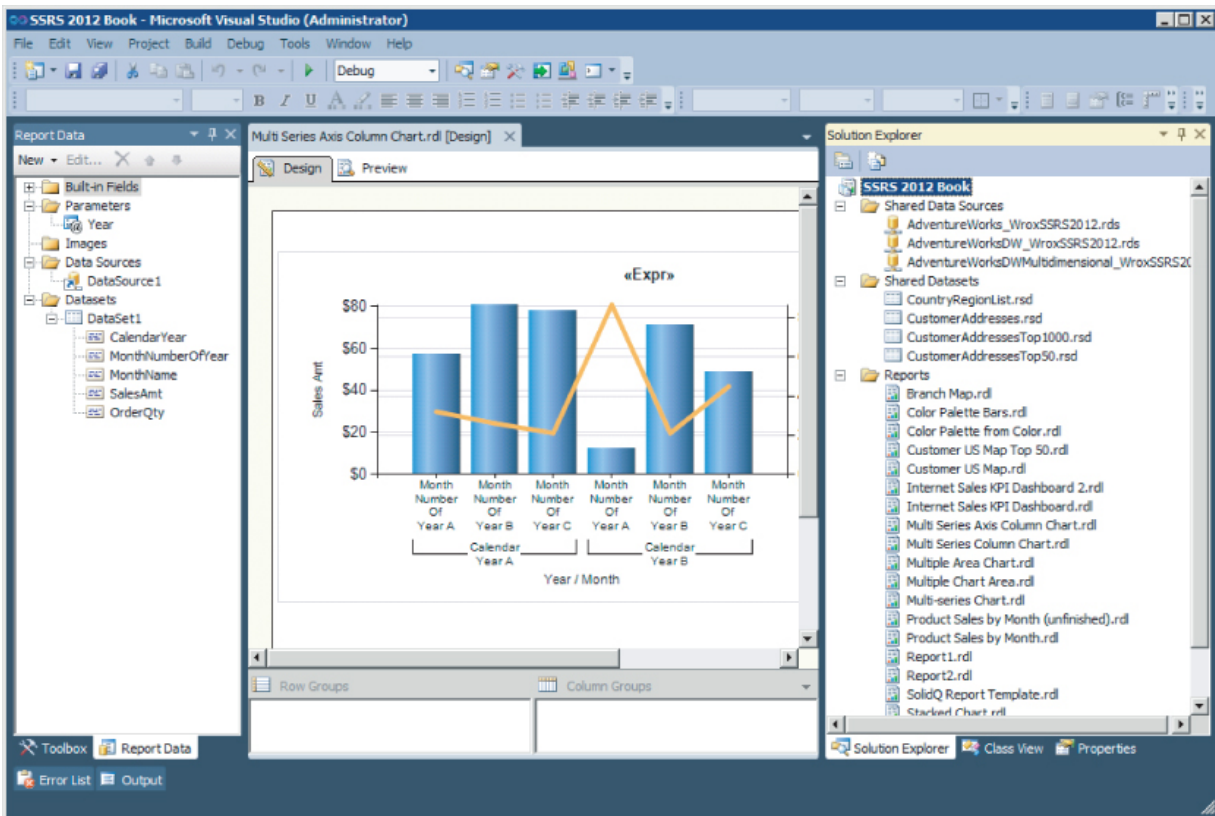
When Reporting Services was first released, the report design experience was optimized for programmers and application developers who were accustomed to using Visual Studio, a product designed to help technically inclined programmers create custom software. When the product team completed the first-generation product, they immediately went to work on a set of tools to enable business users to design their own reports. This offering came to the market in stages with subsequent product releases. A brief understanding of this history will help you better appreciate how we arrived at the current set of tools and capabilities.

For the more technically inclined, the report design experience in the Visual Studio shell, called the SQL Server Data Tools (SSDT), is both familiar and powerful (see [Figure 1.2](#)).



In previous versions of SQL Server, the Visual Studio-based design tool was called “Business Intelligence Development Studio” (or BIDS for short). If you hear one of us SSRS old-timers refer to this tool by the former name, you can assume that we're talking about SSDT. It's going to take me a while to get use to this change and to start using the new name.

**[Figure 1.2](#)**



Simple reports are fairly easy to design, and advanced capabilities are possible using a variety of tools that will make most application developers feel right at home. Like other Visual Studio solutions, report definition files are placed in folders that can be managed as a single deployment unit. Reports, data sources, shared datasets, and all other design elements can be managed with integrated version control in the SSDT shell.

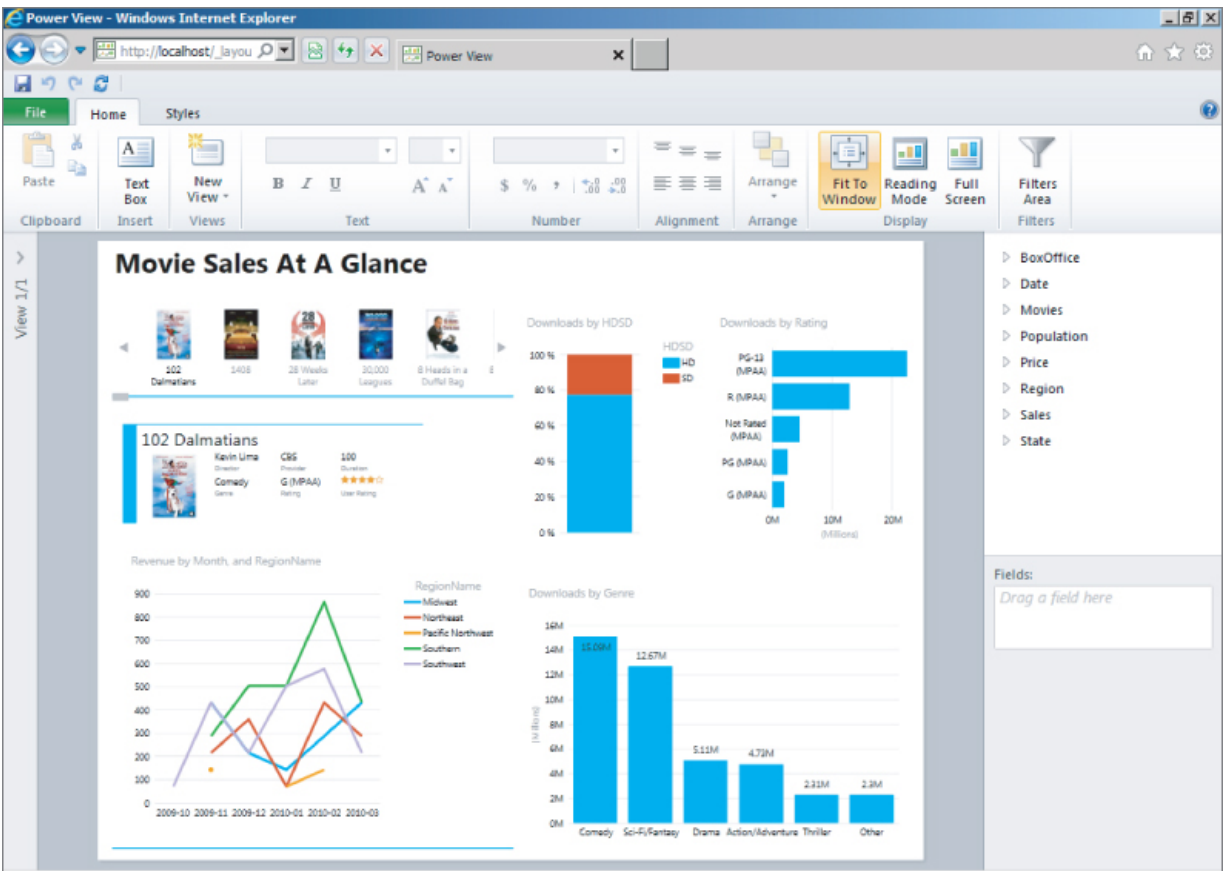
Many technical experts use either Report Builder or SSDT to design their reports. Both tools include a simple Report Wizard that can lead you through designing common reports. Table, grouped, matrix, and chart reports are relatively easy to build just by following the Wizard prompts and setting a few properties.

## User-Designed Reports

SQL Server 2012 brings us full circle with two capable self-service design tools. Self-service reporting has been on the minds of many people in the industry for a long time. For Microsoft, the quest to create the perfect easy-to-use BI tool has produced several different products, each with its own unique capabilities. Under the Reporting Services umbrella, two ad hoc reporting tools serve different needs. The current incarnation of Report Builder is based on the mature report definition architecture we've seen progress over the past eight years. Report Builder reports can span the spectrum from simple to complex, with many design options. The Power View visualization report tool introduces an exciting and dynamic data browsing and exploration experience. Power View reports are based on a tabular semantic data model and are surfaced in an intuitive SharePoint-based interactive designer, as shown in [Figure 1.3](#).

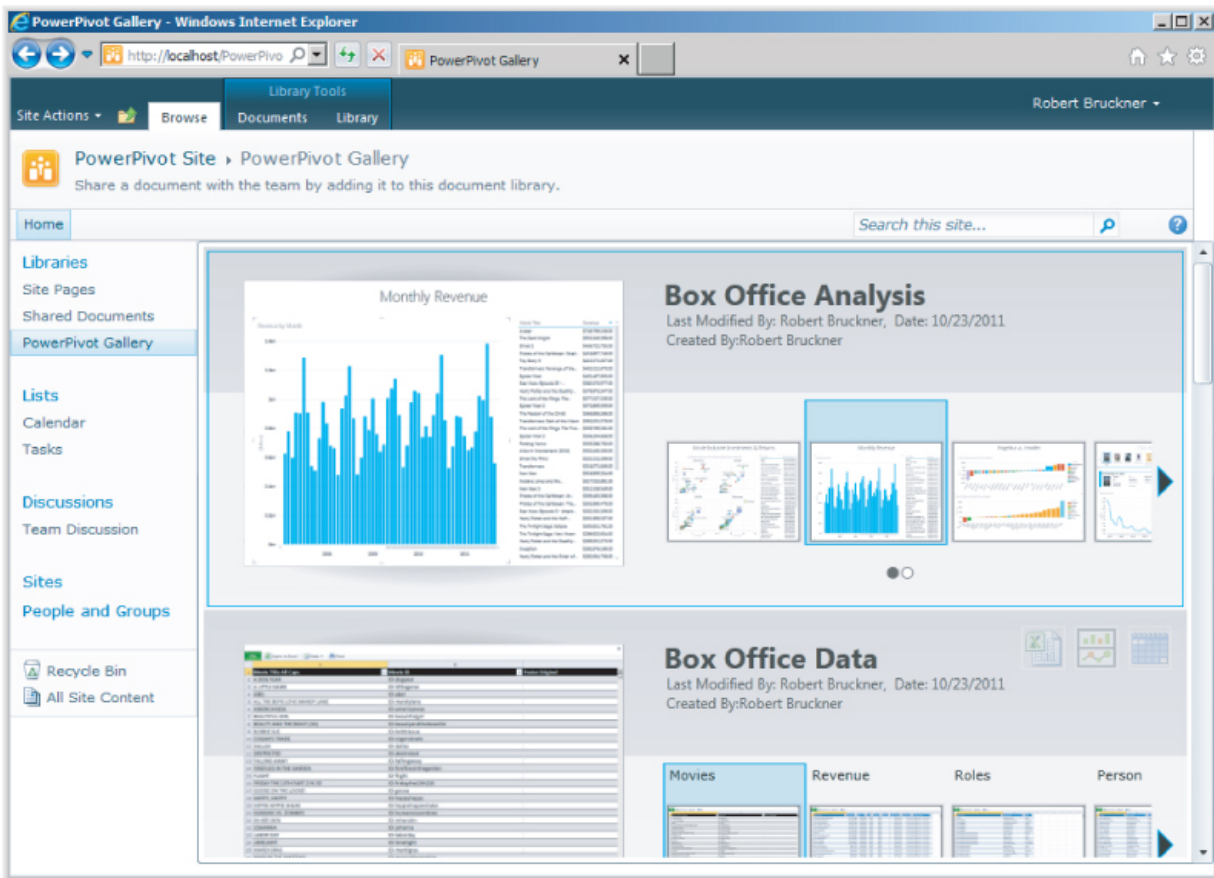
### **[Figure 1.3](#)**





The SharePoint environment provides a rich content management and presentation interface for a variety of reports and content types. [Figure 1.4](#) shows the PowerPivot Gallery, with thumbnail previews of Power View reports and PowerPivot workbooks.

**[Figure 1.4](#)**



Chapters 12 and 13 cover the Power View tool and tabular models for users and for those charged with supporting and enabling this capability in their organization.

The first generation of self-service reporting in SSRS was a step toward the robust capabilities in the current product. Report Builder 1.0 was a basic tool introduced with SSRS 2005 that produced a simple but proprietary report with limited capabilities. It was a great tool for its time that allowed users to simply drag and drop data entities and fields from a semantic data model to produce simple reports. Today, the latest version of Report Builder creates reports that are entirely cross-compatible with SSDT and that can be enhanced with advanced features. Consider Report Builder 1.0 yesterday's news. If you're using it now, I strongly suggest making the transition to the newer tool set.